6J−1

画像処理ワークステーションのためのソフトウェア環境(6)
−画像処理コマンドインタプリタμV-Sugar−

古賀慎一郎　岡崎 洋　田村秀行
キヤノン株式会社 情報システム研究所

1. まえがき

V-Sugarは，**VIEW-Station**ソフト体系[1]の中核をなすプログラミング・ツールで，画像データ型の利用と関数型記述を特徴としている．このV-Sugarの特徴を継承しながら，対話的な画像処理実験を容易にするコマンドインタプリタ**μV-Sugar**を開発した．本稿では，μV-Sugarの設計及び実現について報告する．

2. 設計方針

V-SugarがC++を親言語としたコンパイラ言語であるのに対し，μV-Sugarでは，アルゴリズム記述能力を一部縮小して，インタプリタ型の簡易実行形式を重視した．以下に，V-Sugarと比較しながら，μV-Sugarの特徴を述べる．

(1) 画像データ型の扱い

μV-Sugarでは，V-Sugarに準じ画像処理のためのデータ型を導入する．V-Sugarではコンパイル時にデータ型チェックを行っていたが，μV-Sugarではコマンド投入の度ごとにデータ型チェックを行う．また，V-Sugarでは，変数のデータ型を明示的に宣言するのに対し，μV-Sugarでは関数出力の変数への代入時に型が決定される．

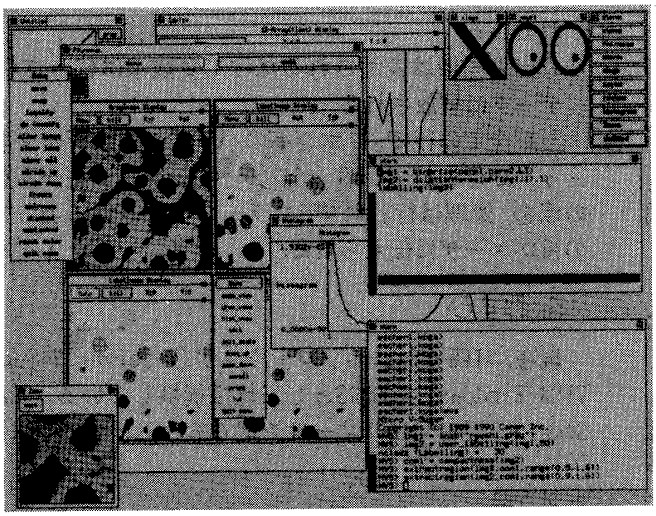

図1 μV-Sugarの操作環境

(2) データの解放機構

多くのメモリを必要とする画像データを扱うため，変数に代入しない中間処理結果を自動的に解放する方式を採用する．例えば，関数形式の入れ子で記述されたコマンドが実行される場合，次の処理への受け渡しのみに利用される中間結果は自動的に解放される．また，変数に代入されているデータに対しても，変数を明示的に削除することにより解放することができる．

(3) コマンド追加機能

既存のコマンドの組み合わせからなる一連の処理をプログラムとしてファイルに記述し，それを1つのコマンドとして登録・実行するスクリプト機能を持たせる．また，V-Sugarで記述した新規の画像処理関数を新たにコマンドとして登録できるようにする．

(4) 表示機能との一体化

μV-Sugarのコマンド実行毎に，処理結果を順次ディスプレイに表示する．画像処理結果の表示部には**VIEW-Windows**[2]を採用しているので，データ型に応じた表示が自動的に選択される．図1に表示機能と一体となったμV-Sugar操作環境を示す．

3. 言語仕様と処理系の構成

μV-Sugarは以下の言語仕様とした．

(a) シンタックス

- V-Sugar上の画像処理ライブラリ**V-SugarLib**のシンタックスに準じ，下に示す関数型の記述を採用する．

  コマンド名(パラメータ1, パラメータ2,…)

- コマンドを関数型にしたことにより，下記のようなコマンドの入れ子も記述可能である．

  コマンド名(パラメータ1, コマンド名(パラメータ2,…),…)

- 処理結果は下に示す代入文により，変数に代入可能である．

  変数＝コマンド名(パラメータ1, パラメータ2,…)

(b) 変数

- 変数は，代入の左辺に現われることで自動的に宣言され，全てのデータ型の値を代入できる．即ち，変数に対してデータ型を明示的に宣言する必要はない．
- スクリプト内の変数はスクリプト内でのみ有効なローカル変数とし，それ以外のものはグローバル変数と

Software Environment of Vision and Image Engineering Work-Station (6)
− μV-Sugar : A Command Interpreter for Image Computing −
Shinichirou KOGA, Hiroshi OKAZAKI, Hideyuki TAMURA
CANON Inc., Information Systems Research Center

表1 μV-Sugarが提供するコマンドの分類

操作場所	コマンド
コマンド インタプリタ ウィンドウ内	・画像処理コマンド binarize (二値化) median (メジアンフィルタ) 等 ・システムコマンド delete (変数の消去) disp (データの表示) setdisp/unsetdisp (表示モードの変更) load/save (データのロード/セーブ)
VIEW-Windows を利用した 画像表示 ウィンドウ内	・メニューコマンド zoom_up (ズーム) hist_view (ヒストグラム表示) 等
UNIXのShellが 動作する ウィンドウ内	vschgcm (新規画像処理コマンドの登録)

する.

(c) スクリプト

・スクリプト呼び出し時にはパラメータの受け渡しが可能で, スクリプト内では変数(parm1, parm2, …)によって参照する.

・最後に実行したコマンドの出力をスクリプトの戻り値とする.

・スクリプト呼び出し時には型チェックは行わず, スクリプト内の各行の実行時に型チェックが行われる.

画像処理コマンド名は, 基本的にはV-SugarLibの関数名を踏襲している. この他に, μV-Sugarインタプリタでは, システムコマンドをいくつか用意している. また, 表示画像に対しては, VIEW-Windowsで用意されているメニューコマンドが利用できる(表1).

図2にμV-Sugarの処理系の構成を示す. μV-Sugar処理系は, V-Sugarで記述されており, コマンドの一行入力に対し, パラメータの型チェックやデフォルトパラメータを補うなどの処理を行った後, V-SugarLib関数を呼び出す.

4. μV-Sugarの動作

図3にμV-Sugarプログラミングの例を示す. 以下, 図中の番号に沿ってμV-Sugarの動作を説明する.

(a1) システムコマンドloadを利用し, ファイル"ryushi.gray"から画像データを入力し, 変数に代入する. この画像は自動的に画面に表示される.

(a2) スクリプトuser_binarizeの実行

(b1) ローカル変数img1は, スクリプト外のimg1とは別のものである. スクリプト呼び出し時のパラメータimg1, 60はスクリプト内のparm1, parm2で参照される. 第3パラメータのLTは, μV-Sugarが提供している定数である.

(b2) コマンドの入れ子として記述されているerosionの出力画像は, dilationの入力画像として受け渡され, この行が実行されるとただちに解放される. (b2)の出力がスクリプトの出力となる.

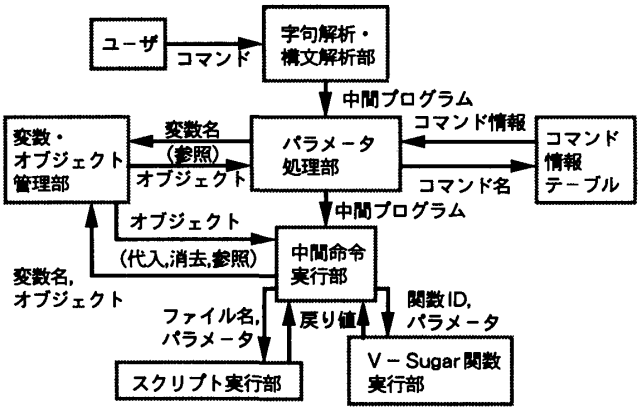

図2 μV-Sugar処理系の構成

```
MVS> img1 = load("ryushi.gray")                    (a1)
MVS> img2 = user_binarize(img1,60)                 (a2)
MVS> img3 = labelling(img2)                        (a3)
MVS> cmp1 = compactness(img3)
MVS> extractregion(img3,cmp1,range(0.9,1.6))       (a4)
```

(a) コマンド列

```
img1 = binarize(parm1,parm2,LT)                    (b1)
dilation(erosion(img1,1),1)                        (b2)
```

スクリプトuser_binarize

(b) スクリプト

図3 μV-Sugarプログラミングの例

(a3) 第2パラメータであるラベリング時の連結数がここでは省略されており, デフォルト値(8連結)が用いられている.

(a4) 出力画像は表示されるだけで変数には代入されず, データは解放される.

上記全ての行で, コマンドのパラメータのデータ型チェックが行われている. パラメータのデータ型が不整合の場合, 処理は行われず, エラーメッセージによってその旨がユーザに伝えられる.

5. むすび

現在, VIEW-Stationソフトウェア(V-Server, V-Sugar, VIEW-Windows)は準PDSとして無償提供している. 本μV-Sugarも, 近日中にリリース予定である.

参考文献

[1] 佐藤,岡崎,河合,山本,田村:"画像処理ワークステーションVIEW-Stationのソフトウェアアーキテクチャ", 情報処理学会論文誌, Vol.31, No.7 (1990).

[2] 河合,岡崎,佐藤,田村:"画像処理ワークステーションとウィンドウシステム", 第20回画像工学コンファレンス, 2-14, pp.87-90 (1989).